CLAIMS

1. A method for secured software patching and upgrade in a distributed wireless sensor network, which comprises:

providing a spanning-tree network of communications nodes with at least one root node and at least one software upgrade repository;

receiving a software upgrade with the root node;

communicating the upgrade from the root node to the software upgrade repository; and

installing the upgrade on the software upgrade repository by authenticating a patch key and delivering and installing the upgrade in the software upgrade repository after authentication occurs.

2. The method according to claim 1, which further comprises providing the communications nodes as sensor devices each sensing, processing, transmitting, receiving, and actuating in a given geographical area.

3. The method according to claim 1, which further comprises:

deploying and managing the patch key of the software upgrade repository with the root node; and

defining a length of the patch key with the root node.

4. The method according to claim 1, which further comprises:

carrying out subgroup controller functions with the software upgrade repository;

coordinating new patch key deployment with the software upgrade repository; and

managing all of the nodes underneath the software upgrade repository on the same branch of the spanning tree with the software upgrade repository.

5. The method according to claim 1, which further comprises varying a length of the patch key on at least one branch of the spanning-tree.

6. The method according to claim 1, which further comprises providing at least one root node of the network as a gateway to another network.

7. The method according to claim 1, which further comprises providing at least one root node of the network as a gateway to the Internet.

8. The method according to claim 1, which further comprises carrying out the installation of the upgrade in parallel on a plurality of software upgrade repositories.

9. The method according to claim 1, which further comprises carrying out the installation of the upgrade in parallel on a plurality of software upgrade repositories within the same upgrade session.

10. The method according to claim 1, which further comprises carrying out the installation of the upgrade in parallel on orthogonal branches of the network.

11. The method according to claim 1, which further comprises carrying out the installation of the upgrade in parallel on orthogonal branches of the network within the same upgrade session.

12. The method according to claim 11, which further comprises forming, with the nodes in the same session, a group with a unique group session key determined by the software upgrade repository.

13. The method according to claim 12, which further comprises starting the software upgrade installation from the software upgrade repository along a respective branch and repeating the software upgrade installation through the branch until all leaf nodes on the branch have the upgrade installed thereon.

14. The method according to claim 13, which further comprises generating a two-byte patch key with the Diffie-Hellman algorithm in every step along the branch on each node in the same session.

15. The method according to claim 14, which further comprises, before carrying out the upgrade, exchanging at least one of a key length, a session key, the patch key, and a prime modulus between the two nodes undertaking the upgrade.

16. The method according to claim 15, which further comprises:

maintaining the session key with the software upgrade repository; and

sharing the session key and the prime modulus with all of the nodes in the same session.

17. The method according to claim 1, which further comprises starting the software upgrade installation from the software upgrade repository along a respective branch and repeating the software upgrade installation through the branch until all leaf nodes on the branch have the upgrade installed thereon.

18. The method according to claim 1, which further comprises generating a two-byte patch key with the Diffie-Hellman algorithm on the node.

19. The method according to claim 1, which further comprises carrying out the authentication with variable-length patch keys having a given length for the software upgrade repository and a shorter length for nodes of the network farther away from the root node than the software upgrade repository.

20. The method according to claim 1, which further comprises carrying out the authentication with different length patch keys, a patch key having a given length for communications between the root node and the software upgrade repository and another patch key having a length shorter than the given length for communications between the software upgrade repository and nodes farther away from the root node than the at least one software upgrade repository.

21. The method according to claim 1, which further comprises carrying out the authentication with different length patch keys, a patch key having a given length for communications on an active branch and a length shorter than the given length for communications on an inactive branch.

22. The method according to claim 1, which further comprises:

defining the software upgrade repository to be immediate children of the root node; and

managing the software upgrade repository with the root node.

23. The method according to claim 1, which further comprises providing the network with one software upgrade repository for each branch.

24. The method according to claim 1, which further comprises carrying out the authentication with patch keys generated locally on each node according to the Diffie-Hellman algorithm.

25. The method according to claim 1, which further comprises sharing a respective patch key on software upgrade repositories on branches with orthogonal updating processes running in parallel.

26. The method according to claim 1, which further comprises:

storing the patch key in both the software upgrade repository and the node to be upgraded; and

determining, with the software upgrade repository, if the patch key received from the node is valid and if so, providing a session key to the node.

27. The method according to claim 16, which further comprises generating and exchanging the patch key and prime modulus by:

first, generating the patch key with the software upgrade repository utilizing the key length, a secret key, and a predefined prime modulus;

second, executing the Diffie-Hellman algorithm with the software upgrade repository to obtain the patch key;

third, sending at least the key length, the patch key, and the prime modulus, to the node to be upgraded;

fourth, picking a random secret number and executing the Diffie-Hellman algorithm with the node to be upgraded to generate a patch key of the node, and sending the patch key of the node to the software upgrade repository;

fifth, authenticating correct reception of the patch key of the node as a condition for the software upgrade repository to authenticate a session key back to the node;

sixth, executing the Diffie-Hellman algorithm with the software upgrade repository upon receiving the patch key of the node to generate a session key;

seventh, authenticating the node to proceed and start the upgrade installation on the node when the node receives the session key.

28. The method according to claim 27, which further comprises:

sending an invitation message with a software version number to the node; and

accepting the invitation with the node and sending an acknowledgement message back to the software upgrade repository containing at least one of the software version number of the node, the session key and a node identification.

29. The method according to claim 28, which further comprises repeating the authentication between the node and at least one subsequent node on the branch.

30. The method according to claim 28, which further comprises repeating the authentication between the node and at least one subsequent node on the branch until all nodes in the branch have executed the installation.

31. The method according to claim 28, which further comprises repeating the authentication between the node and at least one subsequent node on the branch until all nodes in orthogonal branches have executed the installation.

32. The method according to claim 31, which further comprises subsequently repeating the authentication between all nodes on other different orthogonal branches until all nodes have executed the installation, each of the different orthogonal branches having a different session key.

33. The method according to claim 1, which further comprises carrying out the upgrade installation by:

downloading at least one upgrade from an upgrade server and saving the upgrade on a device in the network to be upgraded including at least one of the root node, the software upgrade repository, and a node;

sending information regarding present characteristics of the device to be upgraded to the upgrade server and determining, with the upgrade server, if an upgrade needs to be performed for the device;

receiving, with the device to be upgraded, a response to the information sent from the upgrade server and parsing the response to determine what aspects of the device needs to be upgraded;

selecting an appropriate upgrade with the device to be upgraded, sending a request to the upgrade server to send the appropriate upgrade, and downloading relevant upgrade data; and

saving the upgrade data in the device at a temporary storage sector.

34. The method according to claim 33, which further comprises providing the upgrade server as any device in the network able to transfer the upgrade.

35. The method according to claim 33, which further comprises sending the information through packet data.

36. The method according to claim 35, which further comprises providing the packet data with at least one of the group consisting of a start address, a current block number, a data size, data relevant for the upgrade, a total block number, and a checksum

37. The method according to claim 36, which further comprises sending information including at least one of a serial number, a patch version, and a configuration version.

38. The method according to claim 33, carrying out the upgrade installation by:

switching a node in the network to an upgrade mode at a given time; and

switching the node to a working mode if the temporary storage sector is empty and, if the temporary storage sector is not empty:

determining from the upgrade data in the temporary storage sector a destination sector number in software of the device for the upgrade; and

writing the upgrade data from the temporary storage sector over a data section of the destination sector in the software of the device.

39. The method according to claim 38, which further comprises, if the temporary storage sector is not empty:

storing a portion of the software of the device in the destination sector in a temporary memory of the device;

writing the upgrade data stored in the temporary storage sector over the software portion; and

writing the upgraded software portion into long-term memory of the device.

40. The method according to claim 38, which further comprises carrying out the determination step by determining from the data in the temporary storage sector a destination sector number in the software of the node for the upgrade and the data size and copying all data from the destination sector in the software to temporary memory in the device.

41. The method according to claim 38, which further comprises erasing the upgrade data from the temporary storage sector.

42. The method according to claim 38, which further comprises determining, with the device, if there is another upgrade stored in the temporary storage sector, and:

if another upgrade is not present, switching the node to a working node; and

if another upgrade is present:

determining if the other upgrade has been installed in the device and:

if so, switching the node to a working node; and

if not, repeating the temporary storage sector examination step, the destination sector number determination step, the writing step, and the subsequent upgrade determination step until all upgrades are installed in the device.

43. The method according to claim 42, which further comprises continuously searching for a new upgrade different from a last patch entry already installed in the device.

44. The method according to claim 38, which further comprises switching a node in the network to an upgrade mode immediately after the device is turned on.

45. The method according to claim 38, which further comprises, if the temporary storage sector is not empty, retrieving a start address and a data size from the upgrade data in the temporary storage sector.

5

46. A method for secured software patching and upgrade in a distributed wireless sensor network, which comprises:

providing a spanning-tree network of communications nodes with at least one root node and at least one software upgrade repository;

receiving a software upgrade with the root node;

communicating the upgrade from the root node to the software upgrade repository;

carrying out the installation of the upgrade in parallel on orthogonal branches of the network within the same upgrade session and forming, with the nodes in the same session, a group with a unique group session key determined by the software upgrade repository and maintaining the session key with the software upgrade repository; and

installing the upgrade on the software upgrade repository by:

generating patch keys locally on each node according to the Diffie-Hellman algorithm;

exchanging at least one of a key length, a session key, a patch key, and a prime modulus between the two nodes undertaking the upgrade and sharing the session key and the prime modulus with all of the nodes in the same session;

authenticating a patch key and delivering and installing the upgrade in the software upgrade repository after authentication occurs; and

starting the software upgrade installation from the software upgrade repository along a respective branch and repeating the software upgrade installation through the branch until all leaf nodes on the branch have the upgrade installed thereon.

47. In a spanning-tree network of communications nodes, a communications node, comprising:

a receiver for receiving communications from other communications nodes in a communications range;

a transmitter for sending communications to other communications nodes in said communications range;

a memory storing at least ranging information and a unique identification for describing the node; and

a processor connected to said receiver, to said transmitter, and to said memory, said processor being programmed to:

receiving a software upgrade;

communicate the upgrade to another one of the communications nodes; and

install the upgrade by authenticating a patch key and delivering and installing the upgrade in the node after authentication occurs.

48. A microprocessor programmed to carry out the steps of the method of claim 1.

49. A microprocessor programmed to carry out the steps of the method of claim 46.